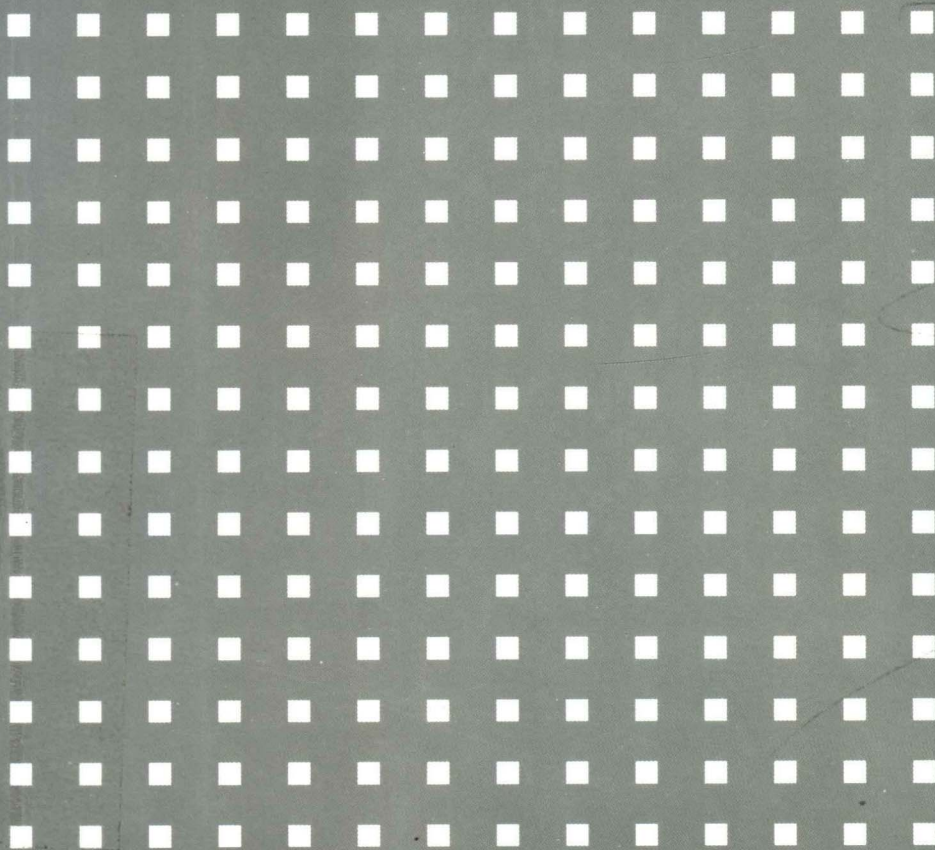


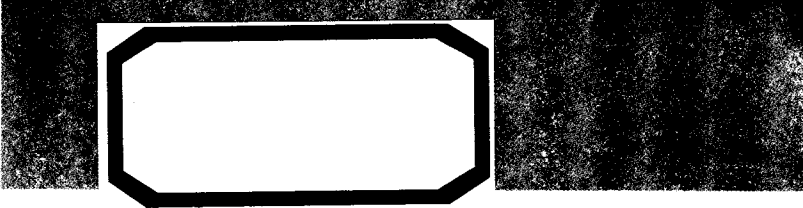
高等学校计算机专业教材精选·图形图像与多媒体技术

计算机图形学实践教程 (Visual C++版)

孔令德 编著



清华大学出版社

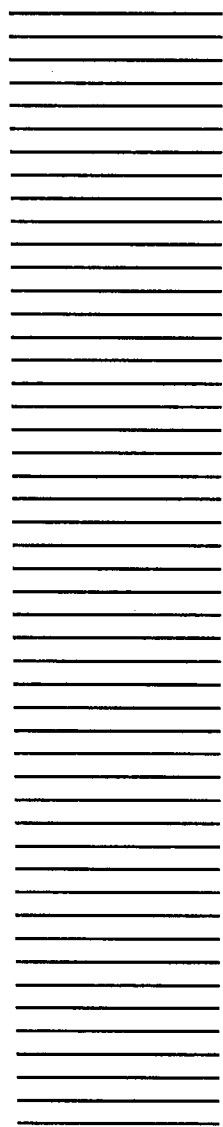


高等学校计算机专业教材精选·图形图像与多媒体技术

计算机图形学实践教程 (Visual C++版)

孔令德 编著

清华大学出版社
北京



内 容 简 介

本书是《计算机图形学基础教程(Visual C++ 版)》(ISBN 978-7-302-17082-2)的配套实践教程。本书的编写原则是将计算机图形学的基本原理教授和具体实现相结合。开发工具选用面向对象 Visual C++ 6.0 的 MFC 框架环境作为开发平台,可以更好地实现真彩色以及对图形的交互式操作。本书共给出 43 个案例,内容包括直线距离加权反走样算法、多边形有效边表填充算法、Bezier 曲线算法和曲面算法、B 样条曲线算法和曲面算法、Sierpinski 海绵算法、分形草文法模型算法、Sierpinski 垫片的 IFS 图形算法、立方体动态隐线算法、立方体动态 Z-Buffer 隐面算法、立方体动态画家隐面算法、明暗处理立方体算法、球的光照模型算法和立方体的纹理映射算法等。

本书的所有案例全部由笔者独立开发,具有自主知识产权。有些算法,如多边形有效边表填充算法、双三次 B 样条曲面算法、Sierpinski 海绵算法、立方体动态 Z-Buffer 隐面算法和球的光照模型算法等内容都是第一次公开发表。

本书算法讲解透彻,算法编写规范,注释清晰,读者可以很容易地按照本书提供的源程序一步一步地完成上机实践。本书相关资源可以从清华大学出版社网站(<http://www.tup.com.cn>)本书相应页面下载。

本书不仅可以作为大学实践教程和高职高专实训教程,还可供从事计算机图形研究的程序员自学使用。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

计算机图形学实践教程: Visual C++ 版/孔令德编著. —北京:清华大学出版社, 2008.5
(高等学校计算机专业教材精选·图形图像与多媒体技术)
ISBN 978-7-302-17148-5

I. 计… II. 孔… III. 计算机图形学—高等学校—教材 IV. TP391.41

中国版本图书馆 CIP 数据核字(2008)第 028129 号

责任编辑:汪汉友

责任校对:焦丽丽

责任印制:何 芊

出版发行:清华大学出版社

<http://www.tup.com.cn>

c-service@tup.tsinghua.edu.cn

社总机:010-62770175

投稿咨询:010-62772015

地 址:北京清华大学学研大厦 A 座

邮 编:100084

邮购热线:010-62786544

客户服务:010-62776969

印 刷 者:北京国马印刷厂

装 订 者:三河市新茂装订有限公司

经 销:全国新华书店

开 本:185×260 印 张:27.75

字 数:676 千字

版 次:2008 年 5 月第 1 版

印 次:2008 年 5 月第 1 次印刷

印 数:1~4000

定 价:39.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。
联系电话:010-62770177 转 3103 产品编号:028992-01

出版说明

我国高等学校计算机教育近年来迅猛发展,应用所学计算机知识解决实际问题,已经成为当代大学生的必备能力。

时代的进步与社会的发展对高等学校计算机教育的质量提出了更高、更新的要求。现在,很多高等学校都在积极探索符合自身特点的教学模式,涌现出一大批非常优秀的精品课程。

为了适应社会的需求,满足计算机教育的发展需要,清华大学出版社在进行了大量调查研究的基础上,组织编写了《高等学校计算机专业教材精选》。本套教材从全国各高校的优秀计算机教材中精挑细选了一批很有代表性且特色鲜明的计算机精品教材,把作者们对各自所授计算机课程的独特理解和先进经验推荐给全国师生。

本系列教材特点如下。

(1) 编写目的明确。本套教材主要面向广大高校的计算机专业学生,使学生通过本套教材,学习计算机科学与技术方面的基本理论和基本知识,接受应用计算机解决实际问题的基本训练。

(2) 注重编写理念。本套教材作者群为各校相应课程的主讲,有一定经验积累,且编写思路清晰,有独特的教学思路和指导思想,其教学经验具有推广价值。本套教材中不乏各类精品课配套教材,并力图努力把不同学校的教学特点反映到每本教材中。

(3) 理论知识与实践相结合。本套教材贯彻从实践中来到实践中去的原则,书中的许多必须掌握的理论都将结合实例来讲,同时注重培养学生分析、解决问题的能力,满足社会用人要求。

(4) 易教易用,合理适当。本套教材编写时注意结合教学实际的课时数,把握教材的篇幅。同时,对一些知识点按教育部教学指导委员会的最新精神进行合理取舍与难易控制。

(5) 注重教材的立体化配套。大多数教材都将配套教师用课件、习题及其解答,学生上机实验指导、教学网站等辅助教学资源,方便教学。

随着本套教材陆续出版,相信能够得到广大读者的认可和支持,为我国计算机教材建设及计算机教学水平的提高,为计算机教育事业的发展做出应有的贡献。

清华大学出版社

前 言

计算机图形学是交互式图形开发的基本理论,同时也是一门实践性的学科。笔者积累十多年的计算机图形学讲授经验,使用 Visual C++ 6.0 的 MFC 框架开发了涉及“基本图形的扫描转换”、“多边形填充”、“二维变换和裁剪”、“三维变换和投影”、“自由曲线和曲面”、“分形几何”、“动态消隐”和“真实感图形”等章节内容的 43 个案例。

本书是《计算机图形学基础教程(Visual C++ 版)》(ISBN 978-7-302-17082-2)的配套实践教程。对于 Visual C++ 的 MFC 框架,本书从使用的角度进行了详细操作说明。本书的程序给出了 *.h 文件和 *.cpp 文件,算法编写规范,注释清晰,读者可以很容易地按照本书提供的源程序一步一步地完成上机实践。

学习完本书,读者可以建立三维场景,对形体施加光照,改变材质或实现纹理映射。在场景中通过鼠标、键盘来控制形体的旋转和动画,基本达到 OpenGL 或 3DS 生成的图形效果。

本书中有许多案例是笔者工作的基础,如有效边表填充算法、透视投影变换、Gouraud 明暗处理、Z-Buffer 消隐算法和光照模型等,希望读者认真体会和理解。

笔者负责主持山西省精品课程“C++ 程序设计”和院级精品课程“计算机图形学”,本书是面向对象语言和计算机图形学原理相结合形成的产物,是笔者十多年教学科研工作成果的总结。

编 者
2008 年 4 月

目 录

| | | |
|-------|-------------------------------|-----|
| 案例 1 | 金刚石图案算法 | 1 |
| 案例 2 | 直线中点 Bresenham 算法 | 14 |
| 案例 3 | 圆中点 Bresenham 算法 | 19 |
| 案例 4 | 椭圆中点 Bresenham 算法 | 25 |
| 案例 5 | 直线距离加权反走样算法 | 32 |
| 案例 6 | 多边形有效边表填充算法 | 38 |
| 案例 7 | 多边形边缘填充算法 | 51 |
| 案例 8 | 区域四邻接点填充算法 | 58 |
| 案例 9 | 二维基本几何变换算法 | 67 |
| 案例 10 | Cohen-Sutherland 直线裁剪算法 | 85 |
| 案例 11 | 中点分割直线裁剪算法 | 94 |
| 案例 12 | 梁友栋—Barsky 直线裁剪算法 | 104 |
| 案例 13 | 三维基本几何变换算法 | 113 |
| 案例 14 | 三视图算法 | 128 |
| 案例 15 | 透视图算法 | 139 |
| 案例 16 | 三次参数样条曲线算法 | 148 |
| 案例 17 | 三次 Hermite 样条曲线算法 | 157 |
| 案例 18 | Bezier 曲线算法 | 164 |
| 案例 19 | 双三次 Bezier 曲面算法 | 172 |
| 案例 20 | 三次 B 样条曲线算法 | 184 |
| 案例 21 | 双三次 B 样条曲面算法 | 193 |
| 案例 22 | 25 个控制点的双三次 B 样条曲面算法 | 204 |
| 案例 23 | Cantor 集算法 | 215 |
| 案例 24 | Koch 曲线算法 | 221 |
| 案例 25 | Peano-Hilbert 曲线算法 | 227 |
| 案例 26 | Sierpinski 垫片算法 | 235 |
| 案例 27 | Sierpinski 地毯算法 | 241 |
| 案例 28 | Sierpinski 海绵算法 | 247 |
| 案例 29 | C 字曲线算法 | 256 |
| 案例 30 | Caley 树算法 | 262 |
| 案例 31 | Koch 曲线文法模型算法 | 269 |
| 案例 32 | 分形草文法模型算法 | 277 |
| 案例 33 | Peano-Hilbert 曲线文法模型算法 | 284 |
| 案例 34 | 分形灌木丛文法模型算法 | 291 |

| | | |
|-------|-------------------------------|-----|
| 案例 35 | Koch 曲线的 IFS 图形算法 | 299 |
| 案例 36 | Sierpinski 垫片的 IFS 图形算法 | 306 |
| 案例 37 | 立方体动态隐线算法 | 312 |
| 案例 38 | 球的动态隐线算法 | 323 |
| 案例 39 | 立方体动态 Z-Buffer 隐面算法 | 334 |
| 案例 40 | 立方体动态画家隐面算法 | 351 |
| 案例 41 | 明暗处理立方体算法 | 369 |
| 案例 42 | 球的光照模型算法 | 388 |
| 案例 43 | 立方体的纹理映射算法 | 421 |
| 参考文献 | | 432 |

案例 1 金刚石图案算法

本案例知识要点

- 在 Visual C++ 6.0 环境下创建 MFC AppWizard(exe)。
- 设计程序模板 Test 的运行界面。
- 修改 TestView.h 和 Testview.cpp 文件。

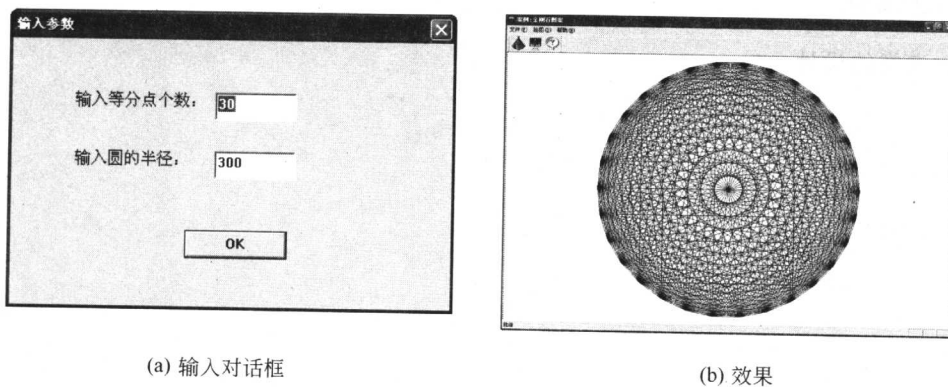
一、案例需求

1. 案例描述

将半径为 r 的圆周 n 等分,然后用直线将各等分点隔点相连,形成的图案称为“金刚石”图案,试编程实现。

2. 案例效果图

案例输入对话框及效果如图 1-1 所示。



(a) 输入对话框

(b) 效果

图 1-1 案例 1 输入框及效果

3. 功能说明

- (1) 程序运行界面提供文件、绘图和帮助这 3 个菜单项。
- (2) 单击绘图菜单,弹出对话框读入圆的等分点个数和圆的半径。
- (3) 以屏幕客户区中心为圆心绘制金刚石图案。

二、案例分析

根据案例需求,需要在 MFC 环境中建立一个由“文件”、“绘图”和“帮助”这 3 个菜单项目组成的菜单,其中“文件”的子菜单为“退出”,完成退出应用程序的工作,“绘图”的子菜单为“Dimond”,用于绘制金刚石图案,“帮助”的子菜单为“关于”,用于显示开发人员信息;定义一个输入对话框类,提供两个参数的输入界面。最后在客户区输出金刚石图案。

本案例设计一个 P2D 类,用于存放各个点的 double 型 (x, y) , 一般而言,最大等分点不会超过 50 个,所以使用 P2D 类定义了大小为 50 的 P2D 类对象数组 $p[50]$ 。

本案例的难点在于内层循环设计时不要进行重复直线连接。例如对于等分点 $n=5$, 连接情况如图 1-2 和表 1-1 所示。

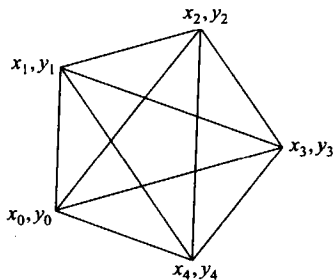


图 1-2 $n=5$ 时的直线连接点

表 1-1 直线连接点的计算

| 起 点 | 终 点 |
|--------------|--|
| (x_0, y_0) | $(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)$ |
| (x_1, y_1) | $(x_2, y_2), (x_3, y_3), (x_4, y_4)$ |
| (x_2, y_2) | $(x_3, y_3), (x_4, y_4)$ |
| (x_3, y_3) | (x_4, y_4) |

为此,设计一个二重循环,代表起点的外层循环从 $i=0$ 循环到 $i=n-2$,代表终点的内层循环从 $j=i+1$ 循环到 $i=n-1$ 。以 $p[i].x, p[i].y$ 作为起点,以 $p[j].x, p[j].y$ 作为终点绘制连接线。

三、案例设计

1. 点类的设计

```
class P2D
{
public:
    P2D();
    virtual ~P2D();
    double x;
    double y;
};
```

为了保证运算精度,本案例设计了 P2D 类,其数据成员为 double 型的 (x, y) 。在类视图 ClassView 里选中 Test Classes 后右击,选择 New Class,打开 New Class 对话框,如图 1-3 所示。在 Class type 中选择 Generic Class 一般类,在 Name 中输入类名 P2D,如图 1-4 所示。单击 OK 按钮,在类视图 ClassView 里添加了新类 P2D,如图 1-5 所示。在文件视图的 Source Files 下 VC++ 向导自动添加了“P2D.h”和“P2D.cpp”文件,如图 1-6 所示。找到“P2D.h”文件双击打开,类的初始定义如图 1-7 所示。修改类的定义,添加 double 型成员变量 x 和 y ,完成 P2D 类的设计,结果如图 1-8 所示。

2. 对话框类的设计

在资源视图 Resource View 上选择 Dialog,右击,选择 Insert Dialog,在对话框内添加静态文本 Static Text 和编辑框 Edit Box 控件并修改属性,Edit1 控件的 ID 为 IDC_EDIT1, Edit2 控件的 ID 为 IDC_EDIT2,对话框的设计结果如图 1-9 所示。这里使用了 VC++ 控件 Controls,可以在工具条空白处右击,选择 Controls 使之显示,如图 1-10 所示。

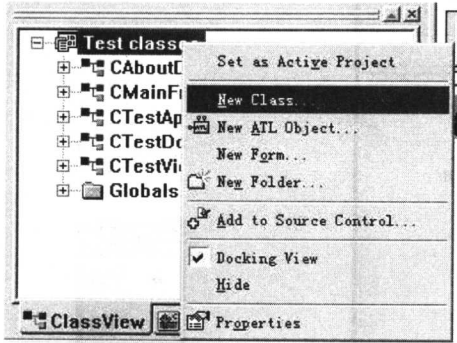


图 1-3 添加新类

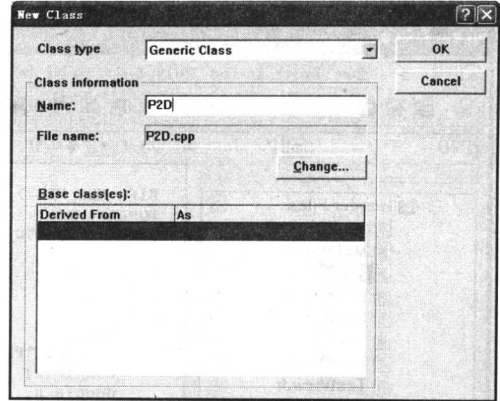


图 1-4 定义新类

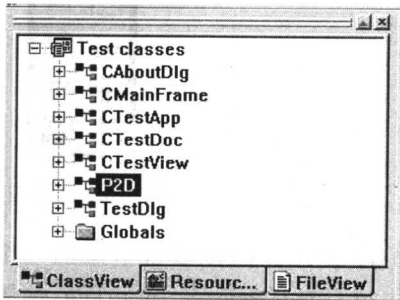


图 1-5 添加的新类 P2D

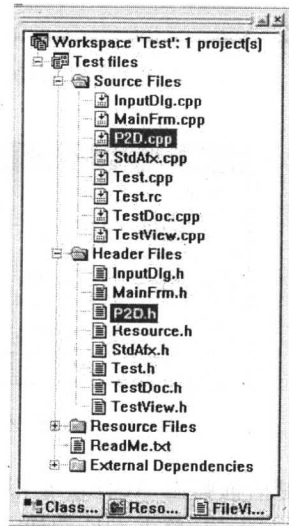


图 1-6 “P2D.h”和“P2D.cpp”文件

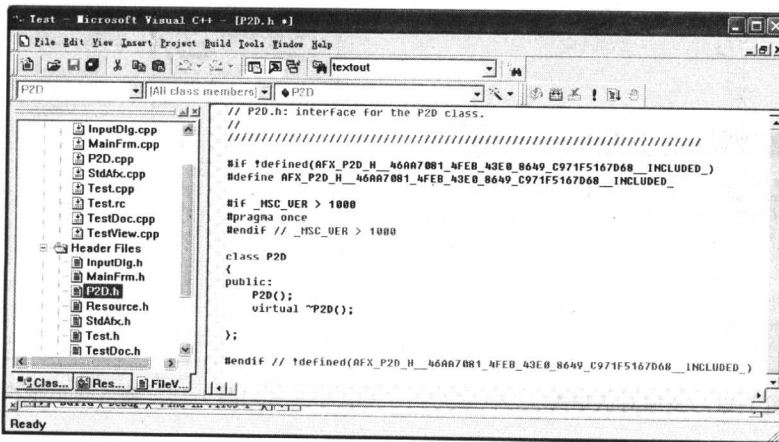


图 1-7 P2D 的初始定义

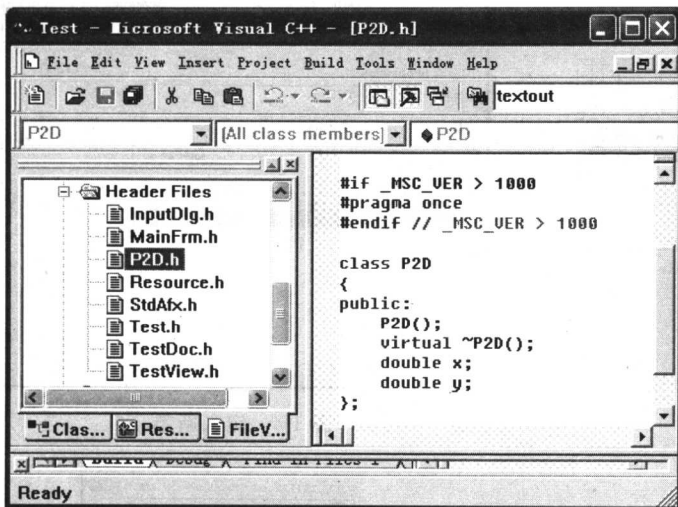


图 1-8 P2D 的完整定义

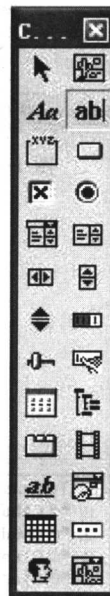


图 1-10 控件列表

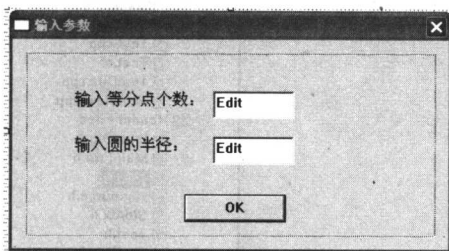


图 1-9 参数输入对话框设计

双击对话框弹出 Adding a Class 对话框,如图 1-11 所示。保持默认值添加一个新类 Create a new class,单击 OK 按钮。在弹出的对话框中填写对话框类名 InputDlg,如图 1-12 所示,单击“确定”按钮,则对话框类添加完毕。

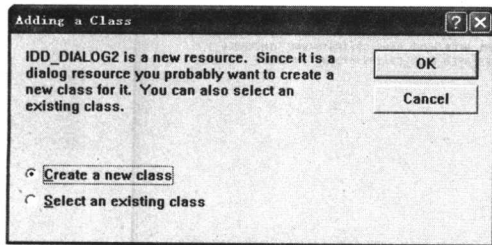


图 1-11 添加对话框类

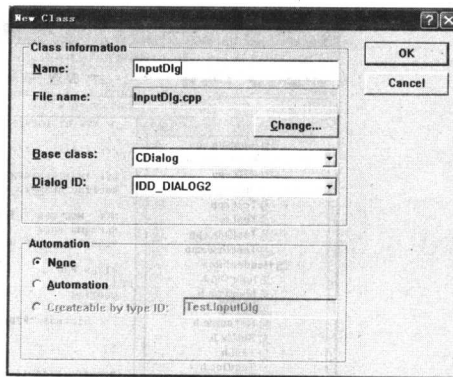


图 1-12 输入对话框类名

下面对对话框类进行设计。在 View 菜单下选择 MFC ClassWizard 菜单项,选中 Member Variables 标签页。修改成员变量的名称和类型。其中 IDC_EDIT1 的成员变量名为 m_n,类型为 double,代表等分点个数,IDC_EDIT2 的成员变量名为 m_r,类型为 double,代表圆的半径,单击 OK 按钮退出。设计结果如图 1-13 所示。

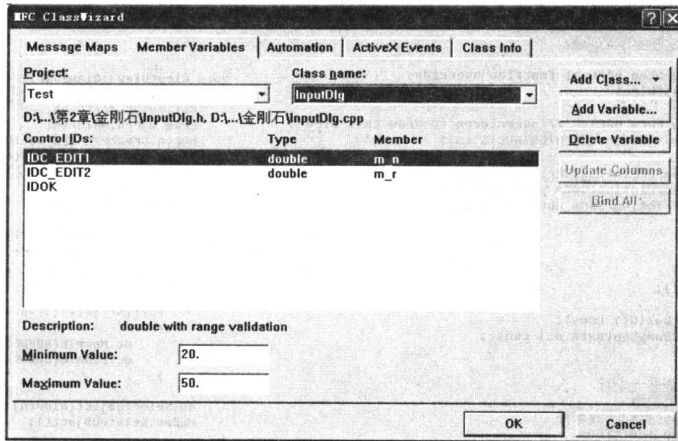


图 1-13 对话框类成员变量的设计

3. CTestView 类的设计

(1) 数据成员。

① double MaxX, MaxY;

声明屏幕的最大 x 坐标和 y 坐标。

② P2D p[50];

声明等分点数组;

③ double n, r;

声明等分点个数和圆的半径。

(2) 成员函数。

① void GetMaxX();

求屏幕的最大 x 坐标函数。

② void GetMaxY();

求屏幕的最大 y 坐标函数。

③ void Diamond();

绘制金刚石图案函数。

为了能直接利用 MFC 已经建立的 CTestView 类,实现金刚石图案的绘制。下面将数据成员和成员函数添加到 CTestView 类。在文件视图的 Source Files 下找到“TestView.h”文件双击打开,分别添加数据成员和成员函数的声明,结果如图 1-14 所示。需要说明的是在“TestView.h”文件中定义了 P2D 类对象数组 p[50],就需要在本文件中包含“P2D.h”头文件,即在文件头加入 #include “P2D.h”语句。同样为了能在“TestView.cpp”里调用对话框类,就需要在“TestView.h”里包含对话框头文件,即在文件头加入 #include

"InputDialog.h" 语句。

在文件视图的 Source Files 下找到 TestView.cpp 文件双击打开,添加成员函数的定义,结果如图 1-15 所示。

```
// Operations
public:
    void GetMaxX(); //计算屏幕坐标中心x函数
    void GetMaxY(); //计算屏幕坐标中心y函数
    void Diamond(); //绘制金刚石函数
// Overrides
// ClassWizard generated virtual function overrides
//{{AFX_VIRTUAL(CTestView)
public:
    virtual void OnDraw(CDC* pDC); // overridden to draw this view
    virtual BOOL PreCreateWindow(CREATESTRUCT& cs);
protected:
    virtual BOOL OnPreparePrinting(CPrintInfo* pInfo);
    virtual void OnBeginPrinting(CDC* pDC, CPrintInfo* pInfo);
    virtual void OnEndPrinting(CDC* pDC, CPrintInfo* pInfo);
//}}AFX_VIRTUAL

// Implementation
public:
    virtual ~CTestView();
#ifdef _DEBUG
    virtual void AssertValid() const;
    virtual void Dump(CDumpContext& dc) const;
#endif
protected:
    int MaxX,MaxY; //屏幕最大坐标
    P2D p[50]; //定义对象数组
    double n,r; //定义等分点和圆的半径
// Generated message map functions
```

图 1-14 添加变量成员和成员函数的声明

```
void CTestView::Diamond() //金刚石函数
{
    //n为等分点的个数, r为圆的半径
    CClientDC dc(this); //设置客户区设备上下文
    CPen MyPen,*OldPen;
    MyPen.CreatePen(PS_SOLID,1,RGB(0,0,255));
    OldPen=dc.SelectObject(&MyPen);
    double Thta; //thta为圆的等分角
    Thta=2*PI/n;
    for(int i=0;i<n;i++)
    {
        p[i].x=r*cos(i*Thta)+MaxX/2;
        p[i].y=r*sin(i*Thta)+MaxY/2;
    }
    for(i=0;i<n-2;i++)
    {
        for(int j=i+1;j<n-1;j++)
        {
            dc.MoveTo(ROUND(p[i].x),ROUND(p[i].y));
            dc.LineTo(ROUND(p[j].x),ROUND(p[j].y));
        }
    }
    dc.SelectObject(OldPen);
    MyPen.DeleteObject();
}
```

图 1-15 添加成员函数的定义

4. 菜单设计

(1) 设置菜单的 ID。在资源视图 ResourceView 上双击 Menu,打开 IDR_MAINFRAME,修改菜单项,结果如图 1-16 所示。



图 1-16 菜单设计结果

设置子菜单“退出”的 ID 为 ID_APP_EXIT,子菜单 Diamond 的 ID 为 ID_MENUDiamond,子菜单“关于”的 ID 为 ID_APP_ABOUT,如图 1-17 所示。

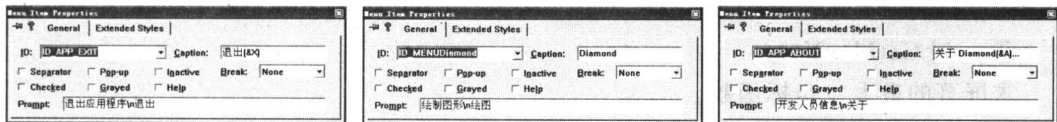


图 1-17 子菜单 ID 设计结果

(2) 添加菜单函数。子菜单“退出”和“关于”的操作函数系统框架建立时已经提供。这里讲解为子菜单 Diamond 添加操作函数的过程。从主菜单 View 中打开类向导 Class wizard。在 Object IDs 中选择 ID_MENUDiamond,在 Class name 中选择 CTestView,在 Message 中选择 Command 后,单击 Add Function 按钮,弹出 Add Member Function 对话框,保持默认菜单成员函数的名字 OnMENUDiamond(),单击 OK 按钮,则在 Member function 中为菜单 Diamond 添加了操作函数 OnMENUDiamond(),该函数是 CTestView

类的成员函数，系统已经自动进行了声明，添加过程如图 1-18 所示。单击 Edit Code 按钮可以对 OnMENUMenuDiamond() 函数进行编辑。

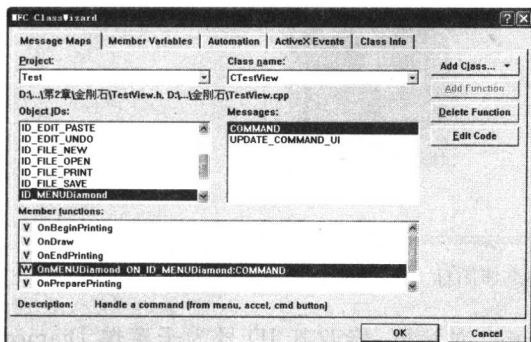


图 1-18 添加菜单的 Diamond 成员函数

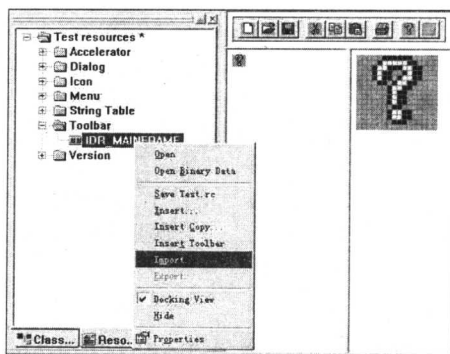


图 1-19 添加图标

5. 工具条设计

在资源视图 ResourceView 中，双击 Toolbar 下的 IDR_MAINFRAME，打开系统框架提供的工具条，将原有图标全部拖动至图标编辑处予以删除。选中 IDR_MAINFRAME，右击，在弹出菜单中选择 Import...，弹出导入资源对话框如图 1-19 和图 1-20 所示。图标导入结果如图 1-21 所示。选中图示 draw.ico 图标，单击 Edit|Copy 菜单。双击 Toolbar 下的

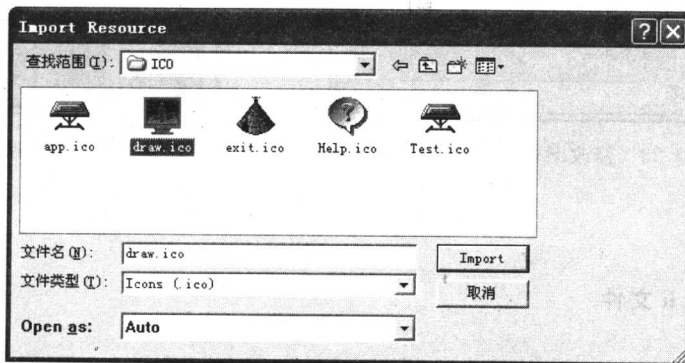


图 1-20 导入资源

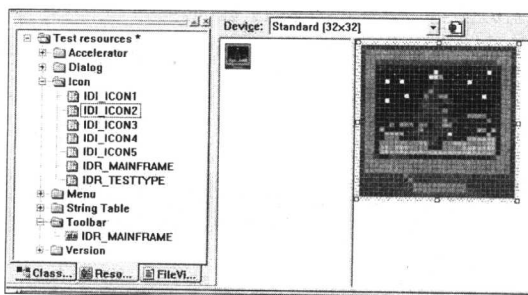


图 1-21 图标导入结果

IDR_MAINFRAME, 打开工具条, 选中空白图标进行 Paste, 结果如图 1-22 所示。

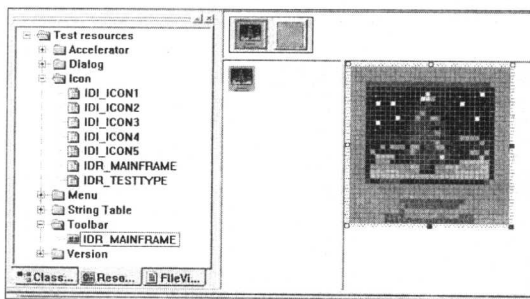


图 1-22 添加图标

双击该图标, 弹出 *Toolbar Button Properties* 对话框, 修改其 ID 号为子菜单 *Diamond* 的 ID, 即 *ID_MENUDiamond*, 以实现图标和子菜单的关联, 如图 1-23 所示。按照以上步骤, 重复设计与子菜单“退出”相关联的图标。工具条设计结果如图 1-24 所示, 其中第一个图标代表“退出”, 其 ID 等于“文件”|“退出”菜单的 ID, 即 *ID_APP_EXIT*; 第二个图标代表“绘图”, 其 ID 等于“绘图”|“Diamond”的 ID, 即 *ID_MENUDiamond*; 第三个图标代表“关于”, 其 ID 等于“帮助”|“关于”菜单的 ID, 即 *ID_APP_ABOUT*。

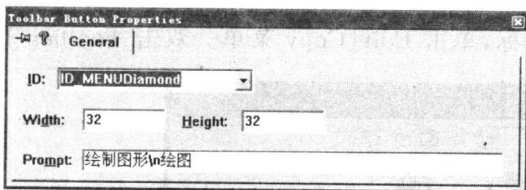


图 1-23 修改图标的 ID



图 1-24 工具条设计结果

四、案例实现

1. TestView.h 文件

代码如下:

```
// TestView.h : interface of the CTestView class
////////////////////////////////////////////////////////////////////
#ifndef AFX_TESTVIEW_H_6DC176FC_B545_4EDA_9B99_E1F77854170A_INCLUDED_
#define AFX_TESTVIEW_H_6DC176FC_B545_4EDA_9B99_E1F77854170A_INCLUDED_
#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000
#include "P2D.h" //包含二维点头文件
#include "InputDlg.h" //包含对话框头文件
class CTestView : public CView
{
protected: // create from serialization only
    CTestView();
```

```

    DECLARE_DYNCREATE(CTestView)
// Attributes
public:
    CTestDoc * GetDocument();
// Operations
public:
    void GetMaxX();           //计算屏幕坐标中心 x 函数
    void GetMaxY();           //计算屏幕坐标中心 y 函数
    void Diamond();           //绘制金刚石函数
// Overrides
    // ClassWizard generated virtual function overrides
    //{AFX_VIRTUAL(CTestView)
public:
    virtual void OnDraw(CDC * pDC); //overridden to draw this view
    virtual BOOL PreCreateWindow(CREATESTRUCT& cs);
protected:
    virtual BOOL OnPreparePrinting(CPrintInfo * pInfo);
    virtual void OnBeginPrinting(CDC * pDC,CPrintInfo * pInfo);
    virtual void OnEndPrinting(CDC * pDC,CPrintInfo * pInfo);
    //}AFX_VIRTUAL
// Implementation
public:
    virtual ~CTestView();
#ifdef _DEBUG
    virtual void AssertValid() const;
    virtual void Dump(CDumpContext& dc) const;
#endif
protected:
    int MaxX,MaxY;           //屏幕最大坐标
    P2D p[50];               //定义对象数组
    double n,r;              //定义等分点和圆的半径
// Generated message map functions
protected:
    //{AFX_MSG(CTestView)
    afx_msg void OnMENUDiamond();
    //}AFX_MSG
    DECLARE_MESSAGE_MAP()
};
#ifdef _DEBUG // debug version in TestView.cpp
inline CTestDoc * CTestView::GetDocument()
    { return (CTestDoc * )m_pDocument;}
#endif
////////////////////////////////////
//{{AFX_INSERT_LOCATION}}
// Microsoft Visual C++ will insert additional declarations immediately before the

```


previous line.

```
#endif // !defined(AFX_TESTVIEW_H__6DC176FC_B545_4EDA_9B99_E1F77854170A__INCLUDED_)
```

2. TestView.cpp 文件

代码如下：

```
// TestView.cpp : implementation of the CTestView class
#include "stdafx.h"
#include "Test.h"
#include "TestDoc.h"
#include "TestView.h"
#define ROUND(a) int(a+0.5) //四舍五入
#include "math.h" //数学头文件
#define PI 3.1415926 //PI的宏定义
#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif //////////////////////////////////////
// CTestView
IMPLEMENT_DYNCREATE(CTestView,CView)
BEGIN_MESSAGE_MAP(CTestView,CView)
   //{{AFX_MSG_MAP(CTestView)
    ON_COMMAND(ID_MENUDiamond,OnMENUDiamond)
    //}}AFX_MSG_MAP
    // Standard printing commands
    ON_COMMAND(ID_FILE_PRINT,CView::OnFilePrint)
    ON_COMMAND(ID_FILE_PRINT_DIRECT,CView::OnFilePrint)
    ON_COMMAND(ID_FILE_PRINT_PREVIEW,CView::OnFilePrintPreview)
END_MESSAGE_MAP()
////////////////////////////////////
// CTestView construction/destruction
CTestView::CTestView()
{
    // TODO: add construction code here
}
CTestView::~CTestView()
{
}
BOOL CTestView::PreCreateWindow(CREATESTRUCT& cs)
{
    // TODO: Modify the Window class or styles here by modifying
    // the CREATESTRUCT cs
    return CView::PreCreateWindow(cs);
}
```